WHAT IS CLAIMED IS:

1. A request-response based transactional auditing method for providing a centralized transactional real-time adaptive identity-driven audit trail over a processing device or sequence of processing devices connected by wired or wireless networks, said method comprising the steps of:
    1) selecting a communication protocol for transmitting a message from a first processing device to a second processing device in a transaction beginning with an incoming request and ending with an outgoing response to the incoming request
    2) creating at least one audit-request object having audit data;
    3) embedding the audit-request object into the incoming request when the incoming request is in compliance with the communication protocol used downstream during a request process;
    4) creating at least one audit-response object at the start of a response process, said audit-response object having audit data;
    5) embedding an audit-response object in an outgoing response, said outgoing response being in compliance with the communication protocol upstream during the response process;
    6) moving the request and the response through a transaction;
2. The method as set forth in claim 1, further comprising the steps of:
    1) creating a unique Transaction ID for said transaction;
    2) adding the Transaction ID to the audit-request object;
    3) verifying the audit-response object contains the Transaction ID;
    4) altering at least part of the audit data carried in the audit-request object or audit-response object at desired points during the transaction;
    5) saving at least part of the audit data which contains at least the transaction id in the audit-request object and the audit-response object in persistence storage at desired points of the transaction, and
    6) removing the audit-response object from the outgoing response object before the outgoing response object leaves the transaction entry point,

whereby the Transaction ID, audit-request object and audit-response object may be used to identify the transaction and log the events during the transaction at any point in order to provide a centralized transactional real-time adaptive identity-driven audit trail that enables the collection and removal of the event trail log for the transaction, and allowing persistence of the audit data of the audit-request object and audit-response object at the centralized user trail repository.

3. The method as set forth in claim 2, further comprising the step of representing the audit-request object and the audit-response object by XML.
4. The method as set forth in claim 3, further comprising the step of:
   1) keeping the audit data of the audit-request object and the audit-response object in an encrypted form during the transaction so that the audit data can be securely transmitted when desired.
5. The method as set forth in claim 3, further comprising the steps of:
   1) passing downstream said audit-request object by means of at least one HTTP header; and
   2) passing upstream said audit-response object by means of at least one HTTP header.
6. In addition to method as set forth in claim 5, said audit-data contains:
   1) user Id

   the user id is a unique identifier that identifies the user. This field uniquely links a user to its activities across the enterprise.
7. In addition to method as set forth in claim 5, said audit-data contains:
   1) session Id

   the session id uniquely identify a user session as shown in the art
8. In addition to method as set forth in claim 5, said audit-data contains one or more fields selected from the group consisting of:
   1) user registry domain
   2) remote connection host
   3) remote connection ip address
   4) roles assigned to the user
   5) any other user related information like user account number

   A user profile that contains any user related information such as contact information, account number associate the said user to other business information.
9. In addition to method as set forth in claim 5, the audit-data contains
   1) TGT which contains:
      a. transaction id

b. user id,

c. token expiration time

d. authorized roles for the user.

10. In addition to method as set forth in claim 5, the audit-data contains one or more fields selected from the group consisting of policy fields:

1) resource that needs to be protected

2) roles that specify if the user is allowed or denied for access this resource

3) rules that associate the user with the resource and the roles to decide who can access the said resource

11. In addition to method as set forth in claim 5, audit-data contains one or more fields selected from the group consisting of audit trail fields:

1) service id

2) authorization status

3) failed reason

4) accessing role

5) service accessing time

6) log-info: Business specific information that need to be collected.

The user based audit-trail which contains any business information for the serviced accessed by the user which can be persisted at the persistence storage for future use such as audit analysis, usage based billing, compliance of the regulations such as Sarbanes Oxley and HIPPA, relieve the corporation liability by providing user activity trail as proof, and the information can also be retrieved from persistence storage for disaster recovery and as source for enterprise data replica.

12. In addition to method as set forth in claim 5, said audit-data contains one or more fields selected from the group consisting of lifespan fields:

1) Transaction Creation Time

2) Transaction expiration-time

3) Transaction time out

Those lifespan fields can help organizations to do performance analysis, setup global time out period to ensure global transaction time out integrity.

13. The method as set forth in claim 5, further comprising the steps of:

1) loading security policies into memory in cached form.

14. The method as set forth in claim 13, further comprising the steps of:

1) adding the policies associated with the resources requested for access to audit-data;

15. The method as set forth in claim 14, further comprising the steps of:

1) authenticating a user, and

2) authorizing a user for a given resource

The authentication and authorization process mentioned here are based on the well-known Kerberos Single-Sign-On System that is applicable to single-domain or multi-domain scenarios.

16. The method as set forth in claim 15, further comprising the steps of:

1) authenticating a user in said first device against user registry;

2) adding authentication event log data into audit-data;

3) creating a encrypted TGT token of said user;

4) adding TGT token into audit-data;

5) forwarding the audit-data containing TGT token to the said first and then said second processing device handling authorization process, said audit-data containing said TGT token facilitating processing of subsequent requests.

6) returning TGT token to the requesting-client as a ticket for the next time access with the associated session id,

17. The method as set forth in claim 16, further comprising the steps of:

1) obtaining TGT token from audit-data,

2) decrypting TGT token to verify the user and retrieve the followings:

a. user id,

b. token expiration time,

c. roles assigned to said user,

d. other relevant information stored within TGT token.

3) retrieving the policies associated with the resources for access from either policy store, in memory cache or audit-data;

4) authorizing said user based on said TGT token and said policies retrieved,

5) adding the authorization event log into said audit data,

6) granting or denying resources access based on authorization status.

7) forwarding the audit-data containing TGT token said second processing device handling authorization process, said audit-data

containing said TGT token facilitating processing of subsequent requests.

18. The method as recited in claim 14, wherein said first processing device is a proxy server and wherein said method further comprises the step of:

1) adding a module to said proxy server, the module adapted to allow dynamic loading of an in-bound filter and an out-bound filter that enable the proxy server to process a HTTP request before sending the request to a resource and to process a response before returning the response.

19. The method as set forth in claim 18, further comprising the steps of:

1) obtaining user profile for a user .

20. The method as set forth in claim 19, further comprising the steps of:

1) authenticating said user based on authentication and authorization API by a API vendor, and

2) authorizing said user for a given resource based on said API

The authentication and authorization process mentioned here are based on the well-known Kerberos Single-Sign-On System that is applicable to single-domain or multi-domain scenarios and whereby vendor independent security infrastructure is achieved across the enterprise.

21. The method as set forth in claim 18, further comprising the steps of:

1) validating data in the request.

2) checking data integrity.

22. The method as set forth in claim 18, further comprising the steps of:

1) performing xml transformation

23. The method as set forth in claim 18, wherein said first processing device and said second processing device are connected via the internet so that web services are transfer requests and responses between said first processing device and said second processing device, and wherein the audit data is added to the audit-request object and the audit-response object regarding activities of web services by the identify of the user and not by the identity of the web services

24. The method as set forth in claim 23, further comprising the steps of:

1) inserting said audit-data via SOAP message,

2) converting SOAP message form of audit-data from response to other forms and moving upstream via HTTP header